Cassie Schaaf
CMPT 220
January 31, 2017

SOFTWARE DEVELOPMENT VS. AGILE DEVELOPMENT

In comparing the software development process from the text, INTRODUCTION TO JAVA PROGRAMMING by Y. Daniel Liang, to the Agile Development process, the differences can be subtle but have an important impact on the final product.

The traditional software development process (SDP) follows a largely linear design. The process begins with specifying the requirements, doing a system analysis, creating a system design, implementing the design, testing the product, deploying finished product, and, finally, maintaining the deployed product. There is minimal iteration, largely between “adjacent” processes, for example, returning to the system design if the implementation fails. The steps in this process are large and comprehensive, each step must be completed before moving onto the next.

Agile Development, based on the The Agile Manifesto, is iterative and incremental. The entire process of planning, analyzing, designing, coding, and testing is done all at the same time. Each part is smaller and is reviewed along the way, without needing to be fully realized as a whole. The principle is that designers should adapt to the challenges that result during the process of development instead of attempting to predict the challenges they may encounter. Every new development is tested as it’s created, instead of waiting for the full review of the completed design.

In my opinion, the benefits of Agile Development FAR outweigh those of the standard process from the text. Locating errors and addressing them immediately is time saving, cost-effective, and less risky than trying to address errors long after the developer has past the creation of the error. For example, if a fatal error occurs early in the process, the whole product can be immediately reevaluated and *adapted* before further damage occurs and cash is output. In the standard method, if a fatal error occurs in the *final* product, the developer must spend incredible amounts of time pouring through code to find the error and risks possibly needing to scrap the full product and start again. For my own design, I've adopted the Agile approach. As a beginner, I feel this is the ONLY way to learn as I go along.